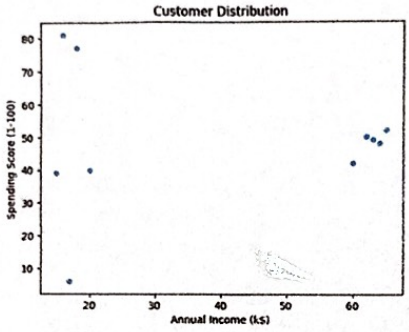
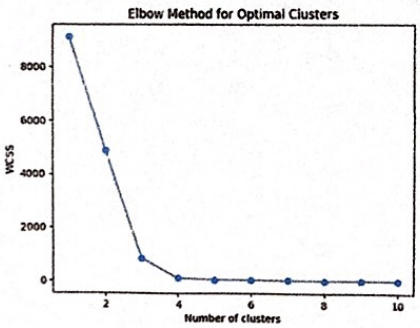
9

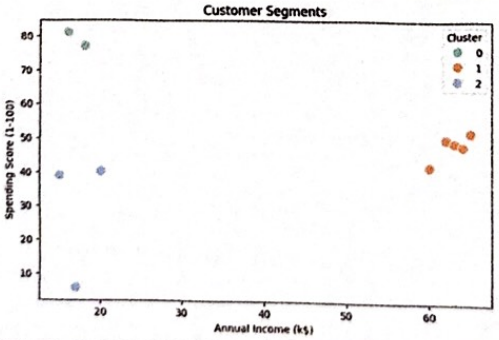
```
import pandas as pd
import matplotlib.pyplot as plt
import Seaborn as sns
from sklearn.cluster import KMeans

# Load or simulate customer data
# Sample dataset : Annual Income vs spending Score
data = {
  'Customer ID' : [1,2,3,4,5,6,7,8,9,10],
  'Annual Income(K$)': [15,16,17,18,20,60,62,63,64,65],
  'Spending Score (1-100)': [39,81,6,77,40,42,50,49,48,52]
}
df = pd.DataFrame(data)
# select features for clustering
X = df[['Annual Income (K$)', 'spending Score (1-100)']]
#visualize data before clustering
sns.scatterplot(X='Annual Income (K$)', y='Spending Score (1-100)',
  data = df)
plt.title('Customer Distribution')
plt.show()

Use the Elbow Method to find optimal number of clusters
wcss = []
for i in range (1,11):
    ~~kmo~~ Kmeans = KMeans (n_clusters = 1, init = 'K-means++',
                    random_state = 42)
    Kmeans.fit(X)
```

```
wcss.append(kmeans.inertia_)
plt.plot(range(1,11), wcss, marker='o')
plt.title('Elbow Method for optimal Clusters')
plt.xlabel('Number of clusters')
plt.ylabel('wcss')
plt.show()

#Apply K-Means with optimal clusters (eg, 3)
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
df['Cluster'] = kmeans.fit_predict(X)

#visualize clustered groups
plt.figure(figsize=(8,5))
sns.scatterplot(
    x='Annual Income (K$)', y='Spending Score (1-100)',
    hue='cluster', palette='Set2', data=df, s=100
)
plt.title('customer Segments')
plt.show()
```

Output for 9


Customer Distribution
Spending Score (1-100)
Annual Income (k$)


Elbow Method for Optimal Clusters
WCSS
Number of clusters


Customer Segments
Cluster
0
1
2
Spending Score (1-100)
Annual Income (k$)